

Sistemi per il recupero delle informazioni

Gabriele Pozzani

A.A. 2014/2015

**Corso di Laurea Magistrale in
Editoria e Giornalismo**

SRI: tipi di interrogazioni

Le interrogazioni nei SRI (1)

- I diversi SRI possono permettere la formulazione di diversi tipi di query:
 - Booleane
 - Prime ad essere state inventate
 - Più diffuse nei SRI “bibliotecari”
 - Vettoriali
 - Booleane estese
 - Più diffuse nel WEB
 - Fuzzy
 - Probabilistiche
 - Linguaggio naturale

3

Le interrogazioni nei SRI (2)

- Ogni tipo di query è caratterizzato da:
 - Un diverso modo (modello) per rappresentare i documenti e le query
 - Un diverso metodo per ordinare (ranking) i documenti recuperati per presentarli all'utente

4

Le interrogazioni nei SRI (3)

- Solitamente i SRI sono basati sull'uso di “termini di indicizzazione”
 - Sono parole chiave
 - In generale sono le parole che compaiono nei documenti
 - Le interrogazioni sono “composte” di termini d'indicizzazione
 - Termini che si ritengono “principali” nei documenti a cui siamo/potremmo essere interessati

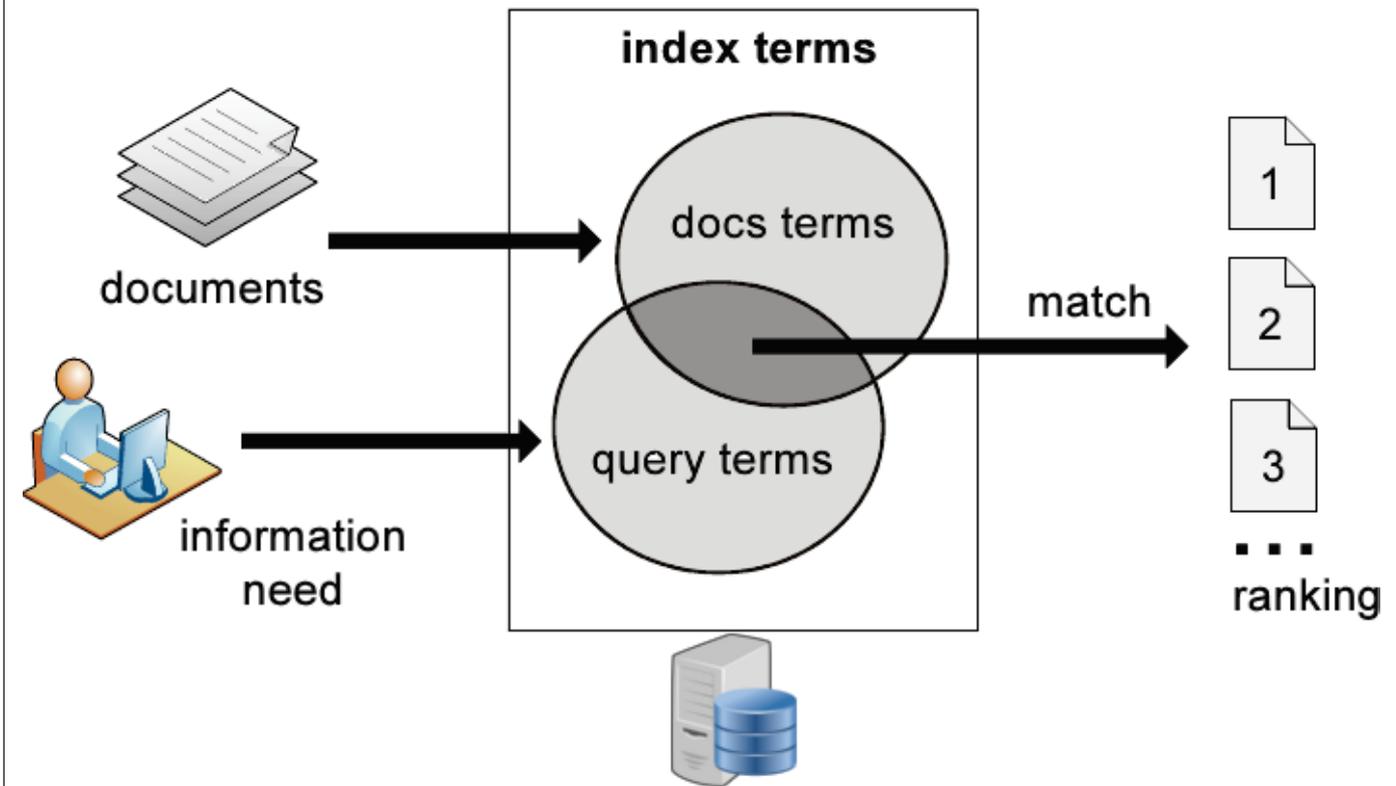
5

Le interrogazioni nei SRI (4)

- Il ranking è l'ordinamento dei documenti recuperati e ritornati all'utente
 - Riflette la rilevanza (pensata dal SRI) rispetto all'interrogazione dell'utente
 - Il SRI cerca quindi di “predire” la rilevanza dei documenti rispetto all'interrogazione dell'utente

6

Processo di recupero delle informazioni



7

Termini d'indicizzazione

- Ogni documento è rappresentato da un insieme di termini d'indicizzazione o parole chiave
- Un termine d'indicizzazione può essere
 - Una parola
 - Una sequenza di parole consecutive in un documento

8

Vocabolario

- L'insieme di tutti i termini d'indicizzazione costituisce il vocabolario
 - $t_1, t_2, t_3, \dots, t_k$
- I termini rappresentano il contenuto del documento
 - Più un termine occorre in un documento più si può presupporre che ne rappresenti il contenuto

9

Rappresentazione matriciale: termine-documento

- L'occorrenza di un termine t_i in un documento d_j mette in relazione t_i con d_j
 - matrice “termine-documento”

$$\begin{array}{c} t_1 \\ t_2 \\ \vdots \\ t_M \end{array} \begin{bmatrix} d_1 & d_2 & \dots & d_N \\ f_{1,1} & f_{1,2} & \dots & f_{1,N} \\ f_{2,1} & f_{2,2} & \dots & f_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ f_{M,1} & f_{M,2} & \dots & f_{M,N} \end{bmatrix}$$

- $f_{k,h}$ rappresenta la frequenza (numero di occorrenze) del termine t_k nel documento d_h

10

Rappresentazione matriciale: esempio

D_1 : “Vivere bene è meglio che vivere.”

D_2 : “Il problema non è vivere a lungo. È vivere bene.”

– matrice “termine-documento”

	D_1	D_2
vivere	2	2
bene	1	1
meglio	1	0
problema	0	1
lungo	0	1

Interrogazioni Booleane

Le interrogazioni Booleane

- Le interrogazioni booleane sono state le prime inventate
 - Usate nei SRI bibliotecari e per la ricerca all'interno di singoli siti WEB
- Formula booleana: lista di valori booleani (sempre o Veri o Falsi) connessi da operatori booleani
 - and (\wedge)
 - or (\vee)
 - not (\neg)
- In una query, un termine rappresenta la presenza o meno di quel termine nei documenti cui l'utente è interessato
 - Un termine o appare o non appare in un documento, in questo è un “valore booleano”
 - Esempio:
`ristorante \wedge (cinese \vee thailandese) \wedge \neg pizzeria`
documenti in cui compare “ristorante” E (compare “cinese” E/O “thailandese”) E NON compare “pizzeria”

Booleano: relativo ad un sistema fondato su due soli valori contrapposti

13

Operatori booleani

- And (\wedge) [*et*]
 - *A and B*: il risultato è vero se e solo se sono veri sia A che B
- Or (\vee) [*ve*]
 - *A or B*: il risultato è vero se e solo se almeno uno tra A e B è vero. Anche: A e/o B.
- Not (\neg) [*non*]
 - *not A*: il risultato è vero se A è falso e viceversa.
- Xor (\oplus) [*aut*]
 - *A xor B*: il risultato è vero se e solo se solo uno tra A e B è vero. Anche: o A o B

14

Tabelle di verità

- Le tabelle di verità elencano tutte le possibili combinazioni di valori degli operandi e il corrispondente valore del risultato
 - Il Vero si indica anche con l'1
 - Il Falso si indica anche con lo 0

A	B	<i>A and B</i>	<i>A or B</i>	<i>not A</i>	<i>A xor B</i>
0	0	0	0	1	0
0	1	0	1	1	1
1	0	0	1	0	1
1	1	1	1	0	0

15

Recupero dei documenti

- Un documento è recuperato se soddisfa (tutta) l'interrogazione, viene scartato altrimenti
 - Un documento o è rilevante o è non rilevante
 - Nessuna via di mezzo

16

Svantaggi (1)

- Non è possibile pesare i termini per dare loro importanze diverse
 - Un termine è presente o assente
 - Ad esempio non è possibile rispondere alla query “musica di Beethoven, preferibilmente una sonata”
- Altri tipi di interrogazioni cercano di risolvere questo problema

17

Svantaggi (2)

- Non essendo i termini pesati, nemmeno i documenti possono essere valutati più o meno rilevanti
 - Non è possibile alcun tipo di ranking dei documenti recuperati in base a maggiore o minore rilevanza
 - Di solito si ordinano alfabeticamente o in base a qualche altra loro caratteristica (e.g., data di pubblicazione)

18

Svantaggi (3)

- Le richieste vanno trasformate in un'interrogazione booleana, ma l'utente non ha sempre ben chiaro il significato degli operatori booleani
 - Errori nella formulazione delle query
- Spesso l'utente pone l'interrogazione interpretando gli operatori logici in funzione del contesto e dei termini
 - Caffè and Brioche or Muffin
 - Impermeabile and Ombrello or Occhiali da sole
- L'uso delle parentesi riduce il problema
 - Caffè and (Brioche or Muffin)
 - (Impermeabile and Ombrello) or Occhiali da sole

19

Svantaggi (4)

- Sistemi di IR diversi possono usare ordini di precedenza degli operatori booleani diversi
 - Gli operatori vengono valutati nell'ordine not, and, or con precedenza da sinistra a destra per quelli dello stesso tipo
 - Gli operatori vengono valutati seguendo strettamente l'ordine da sinistra a destra senza considerare il tipo di operatore
- Le parentesi vengono valutate come un'unità prima di essere combinati con le parti fuori dalle parentesi
- L'uso delle parentesi risolve il problema

20

Google usa il modello booleano? (I)

- In Google l'interpretazione di una query $[t_1 t_2 t_3 \dots t_k]$ è $[t_1 \text{ AND } t_2 \text{ AND } t_3 \text{ AND } \dots \text{ AND } t_k]$
- (approfondiremo ma...) In alcuni casi però alcuni risultati non contengono un termine t_i perché:
 - Una pagina contiene una variante (morfologica, sinonimica, correzione ortografica) di t_i
 - Query lunghe (k molto grande)
 - Di solito portano a pochi risultati quindi...
 - Vi sono pochissimi risultati
 - Meglio comunque dare qualcosa all'utente
- Google permette di usare anche gli altri operatori Booleani
 - OR *[ristorante cinese OR thai]*
 - NOT *[ristorante -cinese]*

21

Google usa il modello booleano? (II)

- Google fornisce anche altri operatori non Booleani
 - Inclusione di parole simili *[~ristorante]*
 - Ricerca all'interno di un sito *[Olimpiadi site:.gov]*
 - Riempi lo spazio vuoto
*["un * risparmiato è un * guadagnato"]*
- Il modello booleano non ordina in modo particolare i risultati
 - Google invece ordina i risultati (in base ad una qualche misura) dal migliore al peggiore
 - Come fa? Va oltre il semplice modello booleano...e vi andremo anche noi ;)

22

Interrogazioni Vettoriali

Le interrogazioni vettoriali

- Non hanno alcuni limiti delle interrogazioni booleane
- Assegnano dei valori/pesi non binari ai termini nelle query e nei documenti
 - I pesi rappresentano l'importanza del termine nelle query e nei documenti
 - I pesi sono usati per calcolare la similarità tra la query e i documenti
 - Il ranking si ottiene ordinando i documenti in ordine decrescente di grado di similarità

Due pesi, due misure

- I pesi nella query possono essere interpretati in due modi (dipende dal SRI in uso)
 - I pesi che si desidera che i termini nella query abbiano anche nei documenti “rilevanti”
 - Il peso relativo tra i termini

Esempio di query vettoriale

$$q = (\text{ristorante}_2, \text{cinese}_{0.8}, \text{thailandese}_{0.8}, \text{pizzeria}_{0.2})$$

- Prima interpretazione: un doc è tanto più rilevante quanto
 - “ristorante” ha peso 2
 - “cinese” e “thailandese” hanno peso 0.8
 - “pizzeria” ha peso 0.2
- Seconda interpretazione: un doc è tanto più rilevante quanto
 - “ristorante” ha peso 10 volte superiore di “pizzeria” e 2,5 volte superiore di “cinese” e “thailandese”
 - “cinese” e “thailandese” hanno peso 4 volte superiore di “pizzeria”

Ma cosa sono questi pesi?

- I pesi dei termini nei documenti possono essere calcolati con varie formule, la più utilizzata nei sistemi odierni è TF-IDF:
 - TF (Term-Frequency)
 - IDF (Inverse Document Frequency)

27

TF-IDF

$$w_{kh} = tf_{kh} * idf_k = \frac{\# \text{ occorrenze } t_k \text{ in } d_h}{\# \text{ docs in cui } t_k \text{ occorre}}$$

- w_{kh} rappresenta il peso del termine t_k nel documento d_h
- tf_{kh} = numero di occorrenze del termine t_k nel documento d_h
 - più un termine compare in un documento, più è importante per descrivere quel documento
- idf_k = inverso [1 fratto ...] della frequenza del termine t_k nella collezione di documenti [≈ 1 fratto num. di doc in cui appare t_k]
[= $\log(N/n_k)$ dove N è il numero di documenti della collezione e n_k è il numero di documenti che contengono il termine t_k]
 - più sono i documenti in cui appare un termine, meno quel termine è importante (per discriminare i documenti)

28

Documenti e interrogazioni come vettori

- Query e documenti sono rappresentati da vettori
 - $d_j = (w_{1j}, w_{2j}, \dots, w_{Mj})$
 - $q = (w_{1q}, w_{2q}, \dots, w_{Mq})$
- w_{kh} rappresenta il peso del termine t_k nel documento d_h (o nella query q)

Vettore: sequenza
ordinata di valori

29

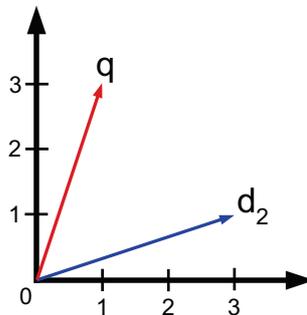
Documenti come vettori

- Ad ogni documento viene associato il vettore contenente i pesi dei termini in esso
 - Supponiamo che il vocabolario contenga solo i termini “ristorante” e “pizzeria”
 - Per ogni termine si calcola il suo peso in ogni documento tramite TF-IDF
 - Supponiamo che il peso di “ristorante” e “pizzeria” in d_1 sia rispettivamente 100 e 300
Supponiamo che il peso di “ristorante” e “pizzeria” in d_2 sia rispettivamente 3 e 1
 - $d_1 = (100, 300)$
 $d_2 = (3, 1)$

30

Similarità tra documenti e query (1)

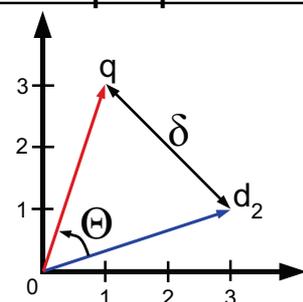
- La similarità tra i documenti e una query corrisponde alla similarità tra i vettori che li rappresentano
 - Ogni vettore può essere rappresentato in uno spazio (sul piano cartesiano nel caso bidimensionale)



31

Similarità tra documenti e query (2)

- La similarità tra i vettori può essere calcolata in diversi modi
 - Distanza euclidea δ : un doc è tanto più simile alla query quanto contiene gli stessi termini con gli stessi pesi
 - 0 \rightarrow massima similarità
 - Misura del coseno Θ : un doc è tanto più simile alla query quanto contiene gli stessi termini in proporzioni simili
 - 1 \rightarrow massima similarità
 - 0 \rightarrow minima similarità



32

Similarità tra documenti e query (3)

- Esempio

$$Q = (1, 3)$$

$$D_1 = (100, 300)$$

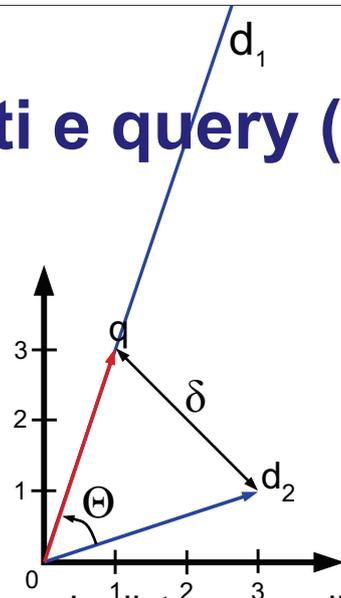
$$D_2 = (3, 1)$$

- $\delta(Q, D_1) = 313,07$

- $\delta(Q, D_2) = 2,83 \rightarrow D_2$ è più simile a Q con la distanza euclidea

- $\Theta(Q, D_1) = 1 \rightarrow D_1$ è più simile a Q con la misura del coseno

- $\Theta(Q, D_2) = 0,6$



33

Similarità tra documenti e query (4)

- L'esempio visto era con soli due termini per permetterne la facile rappresentazione su un piano
- Quando si hanno n termini si hanno vettori con n valori che rappresentano punti in uno spazio a n dimensioni
- La distanza euclidea e la misura del coseno possono essere definite su spazi n -dimensionali
 - Il loro significato comunque rimane lo stesso

34

Recupero dei documenti

- Viene calcolata la similarità di ogni documento con la query
- I documenti sono ritornati in ordine decrescente di similarità
 - Dal più simile al meno simile

35

Google usa il modello vettoriale?

- In Google non possiamo pesare i termini della query 😞
- Facendoci caso in effetti alcuni termini sono più importanti di altri nei singoli documenti/pagine 😊
- I documenti sono ritornati in un ranking in ordine decrescente di “rilevanza” (secondo Google ovviamente) 😊
- Che c'entri il modello vettoriale quindi? (^_-)

Interrogazioni Booleane estese

Facciamo il punto

- Nelle query booleane non è possibile ordinare i documenti recuperati in base ad un ranking
 - Tutti i documenti sono ugualmente importanti per il sistema
- Nelle query vettoriali possiamo solo dire quali termini vogliamo, ma non possiamo indicare alternative (OR) o esclusioni (NOT)

Query booleane estese

- Combiniamo alcune caratteristiche del modello vettoriale con alcune del modello booleano
 - Pesatura dei termini
 - Operatori booleani
 - Possibilità di soddisfare anche solo parzialmente le interrogazioni

39

Idea (1)

- Una query booleana estesa è una normale query booleana
 - $q_{and} = t_1 \wedge t_2$
 - $q_{or} = t_1 \vee t_2$
 - I pesi dei termini della query normalmente non vengono indicati e si suppone siano tutti 1 (ma vi sono SRI che permettono di specificarli)
- I termini di ogni documento sono pesati
 - Usando ad esempio TF-IDF
- I documenti sono recuperati e ritornati in base al peso in essi dei termini nella query
- Ma come si tiene conto degli operatori booleani?

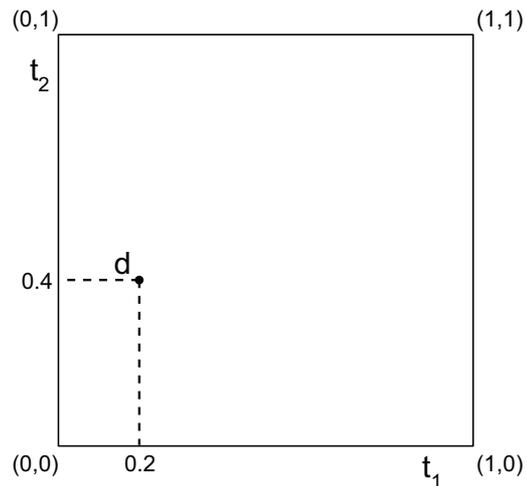
40

Idea (2)

- Poniamo di avere solo due termini t_1 e t_2
 - allora come già visto possiamo rappresentare i documenti su un piano cartesiano

– Esempio, in d

- t_1 pesa 0.2
- t_2 pesa 0.4

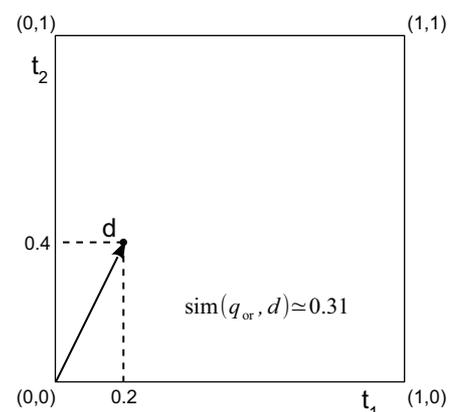


41

Idea (3): or

- Consideriamo la query $q_{or} = t_1 \vee t_2$
 - In un documento d più i due termini sono importanti più d soddisfa la query
 - Prendiamo il punto $(0,0)$ del piano cartesiano come riferimento
 - La distanza del documento d dal punto $(0,0)$ rappresenta la similarità di d con la query q_{or}

$$\text{sim}(q_{or}, d) = \sqrt{\frac{t_1^2 + t_2^2}{2}}$$

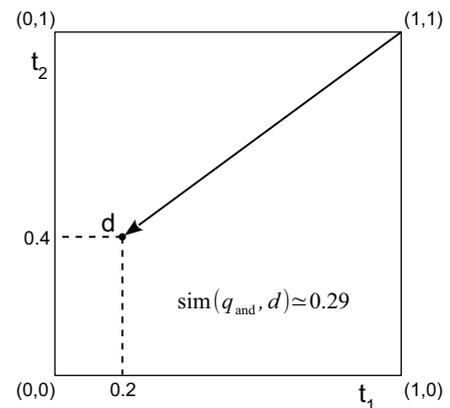


42

Idea (4): and

- Consideriamo la query $q_{and} = t_1 \wedge t_2$
 - La query è soddisfatta al massimo da un documento d quando in d entrambi i termini hanno peso 1
 - Prendiamo il punto $(1,1)$ del piano cartesiano come riferimento
 - La distanza del documento d dal punto $(1,1)$ rappresenta la dissimilarità di d con la query q_{and}

$$\text{sim}(q_{and}, d) = 1 - \sqrt{\frac{(1-t_1)^2 + (1-t_2)^2}{2}}$$



43

Recupero dei documenti

- Viene calcolata la similarità di ogni documento con la query
- I documenti sono ritornati in ordine decrescente di similarità
 - Dal più simile al meno simile

44

Interrogazioni in linguaggio naturale

45

Query in linguaggio naturale (I)

- Frontiera dell'information retrieval
- L'utente può porre un'interrogazione come una domanda nella propria lingua
 - q = “che ore sono?”
- Il sistema cerca di comprendere il significato della domanda e di rispondere
 - Non lista di link a pagine WEB, ma una risposta diretta e puntuale
 - Sistema di “question answering”
- Questo tipo di interrogazione è in generale
 - Imprecisa
 - Inaccuratacome lo è una lingua

46

Query in linguaggio naturale (II)

- Sono il futuro
 - Con la diffusione dei dispositivi mobili e delle ricerche vocali
 - Con l'idea di avvicinare i SRI (e in generale i PC) all'uomo
- Esempi sempre più diffusi sono infatti i sistemi per dispositivi mobili basati sulla voce, tra cui:
 - Google Now
 - Apple Siri

47

Query in linguaggio naturale (III)

- Google Now si basa su
 - Google's user profiling
 - Google Knowledge Graphdi cui parleremo nelle ultime lezioni del corso
- Apple Siri invece si basa su diversi altri servizi tra cui
 - Google Maps
 - Yelp!
 - Google/Bing/Yahoo
 - WolframAlpha

48

WolframAlpha

- "motore computazionale della conoscenza"
[S. Wolfram]
- Elabora input in linguaggio matematico e naturale, fornendo una risposta dettagliata alla domanda
 - Non restituisce una lista di pagine WEB in cui noi dobbiamo cercare la risposta
 - Fornisce "una voce enciclopedica"
- Attualmente è
 - Incentrato soprattutto sulle conoscenze tecniche
 - Solamente in inglese
- <http://www.wolframalpha.com/>